(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
15 juillet 2004 (15.07.2004)

PCT

(10) Numéro de publication internationale
WO 2004/059450 A1

(51) Classification internationale des brevets[7] : G06F 1/00

(21) Numéro de la demande internationale :
PCT/FR2003/003877

(22) Date de dépôt international :
23 décembre 2003 (23.12.2003)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
02/16652 24 décembre 2002 (24.12.2002) FR

(71) Déposants et
(72) Inventeurs : **BANGUI, François** [FR/FR]; 69, rue Dunois, F-75013 Paris (FR). **GONTIER, William** [FR/FR]; 83, rue de Nantes, F-77290 Mitry Mory (FR).

(74) Mandataire : **MORELLE, Guy**; Cabinet Morelle & Bardou, BP 53, F-31527 Ramonville (FR).

(81) États désignés *(national)* : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

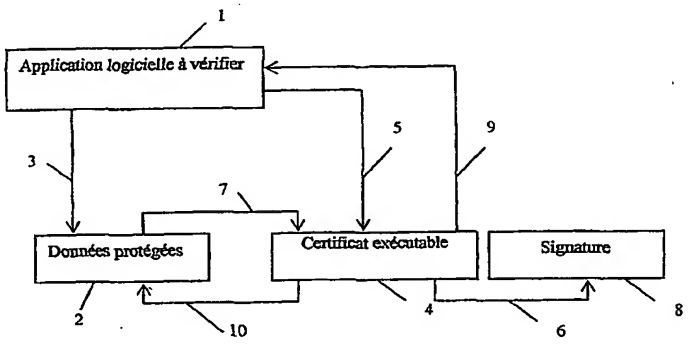
(84) États désignés *(régional)* : brevet ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Publiée :
— *avec rapport de recherche internationale*

[Suite sur la page suivante]

(54) Title: SOFTWARE APPLICATION INTEGRITY VERIFICATION METHOD AND DEVICE

(54) Titre : PROCEDE ET DISPOSITIF DE VERIFICATION DE L'INTEGRITE D'UNE APPLICATION LOGICIELLE



1 SOFTWARE APPLICATION TO BE VERIFIED
2 PROTECTED DATA
4 EXECUTABLE CERTIFICATE
8 SIGNATURE

(57) **Abstract:** The invention relates to a method of verifying the integrity of a software application that can be executed on a host terminal. The inventive method comprises the following steps consisting in: (i) determining at least one series of control instructions forming an executable certificate (4, 15) for the software application, which can be executed by the host terminal during the execution of the software application to be verified (1, 11); (ii) on the host terminal, executing the software application to be verified (1, 11), receiving the executable certificate (4, 15) determined during step (i) and executing the series of control instructions for the certificate which can be executed in the memory context of said host terminal; (iii) comparing the result thus obtained through the execution of the control instructions with the result expected from an authentic software application; and (iv) in the event of a positive comparison, continuing with the execution of the software application to be verified (1, 11).

[Suite sur la page suivante]

WO 2004/059450 A1

— *avant l'expiration du délai prévu pour la modification des revendications, sera republiée si des modifications sont reçues*

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) **Abrégé :** Le procédé de vérification de l'intégrité d'une application logicielle exécutable dans un terminal hôte, comprend les étapes suivantes i) déterminer au moins une suite d'instructions de contrôle formant certificat exécutable (4,15) pour l'application logicielle, exécutable par ledit terminal hôte au cours de l'exécution de l'application logicielle à vérifier (1,11), ii) sur le terminal hôte, exécuter l'application logicielle à vérifier (1,11), recevoir le certificat exécutable (4,15) ainsi déterminé lors de l'étape i) et exécuter la suite d'instructions de contrôle dudit certificat exécutable dans le contexte mémoire dudit terminal hôte, iii) comparer le résultat ainsi obtenu par l'exécution des instructions de contrôle avec le résultat attendu d'une application logicielle authentique et, iv) en cas de comparaison positive, continuer le cours de l'exécution de l'application logicielle à vérifier (1,11).

PROCÉDÉ ET DISPOSITIF DE VÉRIFICATION DE L'INTÉGRITÉ D'UNE APPLICATION LOGICIELLE

La présente invention concerne la vérification de l'authenticité d'une application logicielle exécutée sur un terminal hôte sans nécessairement faire appel à des clés de chiffrement / déchiffrement.

Elle trouve une application générale dans l'authentification d'applications logicielles et plus particulièrement les applications logicielles destinées à être exécutées sur un dispositif de traitement de données, notamment un terminal hôte tel qu'un décodeur de télévision numérique, un équipement de visualisation de contenus multimédia, un micro-ordinateur, une carte à puce, un assistant personnel, une console de jeux, un téléphone mobile ou analogue.

On connaît déjà des moyens d'authentification permettant de vérifier l'authenticité ou l'intégrité d'applications logicielles embarquées et exécutées sur des terminaux hôtes. Généralement, de tels moyens d'authentification mettent en œuvre des fonctions de hachage et/ou des algorithmes cryptographiques qui utilisent des données secrètes telles que des clés privées ou secrètes de chiffrement/déchiffrement cachées dans le logiciel de vérification du terminal hôte. Le plus souvent, ces données secrètes sont protégées par des techniques d'offuscation destinées à rendre plus difficile la rétro-conception.

En pratique, de tels moyens d'authentification embarqués dans les terminaux hôtes sont tout aussi vulnérables que les applications logicielles dont ils sont sensés contrôler l'authenticité. En effet, un pirate averti peut opérer des modifications malveillantes sur ces moyens d'authentification, afin, par exemple, de récupérer les données secrètes, leurrer le système de vérification ou lui faire produire, malgré lui, les résultats attendus.

La présente invention remédie à cet inconvénient. Elle porte sur un procédé de vérification de l'intégrité d'une application logicielle exécutable dans un terminal hôte.

Selon une définition générale de l'invention, le procédé comprend les étapes suivantes :

i) déterminer au moins une suite d'instructions de contrôle formant certificat exécutable pour ladite application logicielle,

ii) sur le terminal hôte, exécuter l'application logicielle à vérifier, recevoir le certificat exécutable ainsi déterminé lors de l'étape i), et exécuter la suite d'instructions de contrôle dudit certificat

exécutable dans le contexte mémoire dudit terminal hôte,

iii) comparer le résultat ainsi obtenu par l'exécution des instructions de contrôle avec le résultat attendu d'une application authentique et,

iv) en cas de comparaison positive, continuer l'exécution de l'application logicielle à vérifier.

On entend ici par le terme "comparaison positive" le fait que toute action, opération, ou modification sur les données utilisées par l'application logicielle à vérifier ou toute action, opération ou modification sur le déroulement de l'exécution de l'application logicielle à vérifier, produise un comportement de l'application logicielle à vérifier identique à celui qui est attendu par le déroulement de l'exécution de l'application authentique.

En renouvelant, à une cadence choisie, la suite d'instructions de contrôle du certificat exécutable, il est possible de mettre en œuvre un nombre important de moyens d'authentification d'une application et il devient quasi impossible de mettre au point des applications logicielles pirates qui déjouent systématiquement le processus de vérification.

Ainsi, le procédé selon l'invention permet de vérifier l'intégrité d'une application logicielle exécutée sur un terminal hôte avec un degré de sécurisation relativement satisfaisant vis-à-vis des pirates adeptes de la rétro-conception et cela sans faire appel à des clés de chiffrement / déchiffrement ou à des composants matériels coûteux.

Selon une réalisation, la suite d'instructions de contrôle est choisie de telle sorte que l'état du contexte mémoire d'une application logicielle authentique après l'exécution de la suite d'instructions de contrôle est identique (sans modification) à l'état du contexte mémoire de l'application logicielle avant l'exécution de la suite d'instructions de contrôle. Ainsi la mise en œuvre du procédé selon l'invention n'apporte pas de dysfonctionnement au niveau du déroulement de l'application logicielle à vérifier si cette dernière est authentique.

Selon une réalisation, dans laquelle le terminal hôte est équipé d'un processeur, la suite d'instructions de contrôle formant certificat exécutable est codée en langage interprétable par ledit processeur du terminal hôte.

En variante, dans laquelle le terminal hôte est équipé d'une machine virtuelle apte à émuler un processeur, la suite d'instructions de contrôle formant certificat exécutable est codée en langage interprétable par la machine virtuelle du terminal hôte.

En pratique, le contexte mémoire d'exécution d'une application logicielle est constitué, entre

autre, des adresses en mémoire des symboles (fonctions, variables,...), des instructions exécutables, des données, et de l'état de la pile d'exécution de l'application. Ces valeurs sont uniques pour chaque application informatique en cours d'exécution et pour chaque type de processeur ou de machine virtuelle.

Dans l'étape i) il est prévu d'établir, dans un environnement sécurisé, une carte du contexte mémoire de l'application logicielle authentique en cours d'exécution, de déterminer, en utilisant les valeurs de cette carte mémoire, une suite d'instructions de contrôle formant certificat exécutable.

Dans l'étape ii), le dispositif d'acheminement du certificat exécutable à destination du terminal hôte est logé dans un circuit électronique de traitement physiquement séparé du terminal hôte. Toujours dans l'étape ii) la récupération des valeurs du contexte mémoire d'exécution se fait par lecture des valeurs aux adresses des différentes zones de mémoire du terminal hôte. Dans ces zones sont logées les instructions exécutables intrinsèques à l'application, les valeurs des variables et les valeurs des références aux fonctions de l'application à contrôler.

A l'étape iii), le résultat obtenu par l'exécution de ladite suite d'instructions de contrôle est une signature de l'application à vérifier. Cette signature est calculée par ladite suite d'instructions de contrôle qui utilise les valeurs du contexte mémoire de l'application logicielle à vérifier en cours d'exécution. De préférence, l'application logicielle comprend des instructions permettant d'insérer et d'exécuter dans son contexte mémoire ladite suite d'instructions en substituant au moins une adresse d'exécution d'une instruction de ladite application logicielle par au moins l'adresse d'une instruction de la suite d'instructions de contrôle formant certificat exécutable.

Selon une autre réalisation, la suite d'instructions de contrôle formant certificat exécutable est transportée dans un flux de données nécessaire à l'exécution de l'application logicielle à vérifier. Pour forcer l'exécution de la suite d'instructions dudit certificat exécutable, lesdites données utiles sont préalablement protégées par une méthode de chiffrement. Le déchiffrement de ces données est correctement effectué par ladite suite d'instructions de contrôle du certificat exécutable si l'application à vérifier est une application authentique. Si le procédé de chiffrement utilise une clé, cette dernière est produite par les instructions de contrôle avec des valeurs du contexte mémoire de l'application logicielle à vérifier, valeurs formant signature de l'application logicielle à vérifier. Les opérations pour obtenir la clé de chiffrement sont codées dans la suite d'instructions du certificat exécutable.

En variante, la méthode de protection est sans clé, la suite d'opérations pour obtenir l'accessibilité

des données est dans la suite d'instructions de contrôle du certificat exécutable. En pratique, la protection des données nécessaires au fonctionnement de l'application logicielle à vérifier est entreprise dans un environnement sécurisé avant que ces données ne soient transmises. La méthode de protection, avec ou sans clé, doit être réversible.

Selon encore une autre réalisation, dans laquelle l'exécution de l'application logicielle fait appel à une carte à puce ou à tout autre circuit sécurisé pour fonctionner, la suite d'instructions de contrôle est logée dans la carte à puce (ou le circuit sécurisé) et envoyée à l'application logicielle à vérifier, l'application logicielle étant apte à récupérer et exécuter ladite suite d'instructions de contrôle ainsi envoyée avec les données dont elle a besoin pour fonctionner. En pratique, l'accès à des données transmises par la carte à puce (ou le circuit sécurisé) doit être nécessaire à l'application logicielle à vérifier pour que celle-ci se comporte de façon identique à une application authentique.

Selon une autre réalisation, à la suite d'une vérification négative de l'intégrité de l'application logicielle à vérifier, le certificat exécutable exécute des instructions faisant appel à des fonctions appartenant à une autre application.

La présente invention a également pour objet un dispositif de vérification de l'intégrité d'une application logicielle pour la mise en œuvre du procédé selon l'invention.

Selon une caractéristique importante de l'invention, le dispositif de vérification comprend des moyens de traitement aptes à déterminer au moins une suite d'instructions de contrôle formant certificat exécutable pour l'application logicielle, exécutable par ledit terminal hôte au cours de l'exécution de l'application logicielle à vérifier, et des moyens de comparaison pour comparer le résultat de l'exécution du certificat exécutable sur le comportement de l'application logicielle à vérifier, avec le résultat attendu du comportement d'une application authentique, et des moyens de modifier l'exécution de l'application logicielle à vérifier en fonction du résultat de la comparaison.

Selon une réalisation, le dispositif de vérification comprend une carte à puce ou tout autre circuit sécurisé apte à contenir d'une part, la suite d'instructions de contrôle formant certificat exécutable et d'autre part, une application réalisant le test de vérification. Le terminal hôte est équipé d'un lecteur de carte à puce (ou d'un moyen de communication avec le circuit sécurisé) et les moyens d'exécution de l'application logicielle à vérifier sont agencés pour charger et exécuter dans son contexte mémoire la suite d'instructions formant certificat. L'application de vérification dans la carte à puce ou le circuit sécurisé est agencée de façon à modifier le déroulement normal

de l'exécution de l'application logicielle à vérifier si le résultat de l'exécution de la suite d'instructions de contrôle n'est pas transmis, dans des conditions définies préalablement, à l'application de vérification dans la carte à puce ou du circuit sécurisé, ou si le résultat de la vérification s'avère négatif.

Selon une variante, le dispositif est apte à déterminer une pluralité de certificats exécutables différents les uns des autres selon une cadence et/ou condition choisie. En pratique, le terminal hôte appartient au groupe formé par les dispositifs de traitement des données, les décodeurs de télévision numérique, les équipements de visualisation de contenus multimédias, les micro-ordinateurs, les cartes à puces, les assistants personnels, les consoles de jeux, les téléphones mobiles ou analogues.

D'autres caractéristiques et avantages de l'invention apparaîtront à la lumière de la description détaillée ci-après et des dessins dans lesquels:

- la figure 1 est une vue schématique illustrant la vérification d'une application logicielle dont les données utiles contiennent le certificat exécutable, selon l'invention, et
- la figure 2 est une vue schématique illustrant la vérification d'une application logicielle utilisant une carte à puce selon l'invention.

En pratique, les termes "intégré" et "authentique" sont ici utilisés indifféremment pour une application logicielle.

On entend ici par "certificat exécutable" une suite d'instructions de contrôle exécutables dans le contexte mémoire d'une application logicielle en cours d'exécution et dont l'exécution produit des effets tels que l'exécution d'une l'application logicielle à vérifier, si cette dernière est intègre, a un comportement identique à celui qui est attendu.

En référence à la figure 1, l'application logicielle à vérifier 1 est embarquée dans un terminal hôte (non représenté). Par exemple, le terminal hôte appartient au groupe formé par les dispositifs de traitement des données, les décodeurs de télévision numérique, les équipements de visualisation de contenus multimédias, les micro-ordinateurs, les cartes à puces, les assistants personnels, les consoles de jeux, les téléphones mobiles ou analogues.

En pratique, l'application logicielle à vérifier 1 traite 3 des données préalablement protégées 2, c'est-à-dire non traitables par l'application à vérifier tant que des instructions de contrôle du certificat exécutable que l'on décrira plus en détail ci-après n'ont pas authentifiées l'application à vérifier. Une méthode pour rendre non accessibles ces données consiste à les chiffrer. Toute

autre méthode de protection réversible est envisageable.

Ces données protégées 2 contiennent 7 un certificat exécutable 4 renfermant une suite d'instructions de contrôle non protégées, qui sont exécutées 5 par l'application à vérifier 1. En pratique, les instructions de contrôle du certificat exécutable 4 sont codées dans le langage du processeur du terminal hôte, encore appelé langage machine. En variante, les instructions du certificat exécutable 4 peuvent aussi être codées dans le langage d'une machine virtuelle, émulant le comportement d'un processeur.

Ces instructions de contrôle du certificat exécutable 4 en langage machine sont des structures binaires préalablement déterminées avant que celles-ci ne soient transmises à l'application logicielle à vérifier.

Les instructions de contrôle du certificat exécutable 4 sont choisies de façon à ce que seule une application logicielle authentique puisse les exécuter pour produire un résultat identique à celui qui est attendu. Comme on le verra plus en détail ci-après, l'absence de dysfonctionnement de l'application logicielle authentique est obtenue en choisissant une suite d'instructions de contrôle de telle sorte que l'état du contexte mémoire d'une l'application logicielle à vérifier 1 après l'exécution de la suite d'instructions de contrôle soit identique à l'état du contexte mémoire de l'application logicielle avant l'exécution de la suite d'instructions de contrôle.

Le certificat exécutable 4 peut aussi être inséré dans le flot de données que l'application 1 est sensée traiter. L'insertion de certificats exécutables dans un flot de données peut correspondre au cas où il est nécessaire d'authentifier une application de traitement de flux multimédia protégé, accessible à l'utilisateur à la condition que ce dernier se soit acquitté des obligations telles que définies par le vendeur des contenus. La source du flux multimédia peut être un point d'émission d'un réseau de diffusion, la mémoire persistante du terminal hôte, ou encore une unité de mémoire extractible du terminal hôte.

Les instructions de contrôle du certificat exécutable 4 sont choisies pour calculer 6 une signature 8 de l'application à contrôler 1, en utilisant 9 le contexte mémoire de l'application à contrôler 1, en cours d'exécution. L'utilisation du contexte mémoire par les instructions de contrôle est réalisé en allant chercher les valeurs de certains symboles (variables, fonctions, instructions exécutables) de l'application à vérifier en cours d'exécution. La récupération de ces valeurs dépend entre autre du modèle mémoire implémenté dans le processeur du terminal hôte.

En pratique, la suite d'instructions de contrôle du certificat exécutable 4 produit 6 une signature

8 qui dépend 9 du contexte mémoire de l'application logicielle à vérifier 1 et utilise cette signature pour lever la protection 10 des données protégées 2. Si l'application logicielle est authentique, les données 2 sont rendues accessibles et leur traitement 3 par l'application logicielle à vérifier 1 produira un résultat identique à celui d'une application authentique.

Dans le cas où le certificat exécutable 4 contenant les instructions de contrôle est inséré dans un flot de données, il est nécessaire de forcer l'application à exécuter ces instructions. Les instructions de contrôle sont alors programmées pour déchiffrer une partie du flot de données que l'application à vérifier doit traiter. Cela nécessite un traitement préalable du flot de données par chiffrement avant que ce flot ne soit accédé, pour traitement, par l'application à vérifier. L'algorithme de déchiffrement peut être implémenté dans les instructions de contrôle ou être disponible sous forme de fonction implémentée dans le terminal, et appelée par les instructions de contrôle. Les clés, si utilisées par l'algorithme de déchiffrement, sont calculées par les instructions de contrôle en utilisant des valeurs du contexte mémoire préalablement définies de l'application en cours d'exécution.

En référence à la figure 2, l'application logicielle à vérifier 11 interagit 12 avec un circuit sécurisé 13, de type carte à puce ou analogue.

La carte à puce 13 transmet 19 un certificat exécutable 15 contenant des instructions de contrôle qui sont chargées et exécutées 16 par l'application logicielle à vérifier 11. Ainsi, pour une application 11 nécessitant 12 une carte à puce 13 pour fonctionner, les instructions de contrôle 15 sont stockées dans la carte à puce 13 et envoyées à l'application à contrôler 11 par le biais du lien interactif 12.

L'application à contrôler 11 utilise 21 pour fonctionner des données 20 qu'elle récupère par interaction 12 avec la carte à puce 13. Ces données 20 contiennent 22 le certificat exécutable 15. Les instructions de contrôle du certificat exécutable 15, lorsqu'elles sont stockées sur la carte à puce 13, sont chargées par l'application à contrôler 11 de façon à ce que celle-ci les exécute 16 selon le principe exposé en référence à la figure 1.

Une signature 18 de l'application logicielle à vérifier est produite 17 par les instructions de contrôle du certificat exécutable en utilisant 14 le contexte mémoire de l'application logicielle à contrôler.

Les instructions de contrôle du certificat exécutable interagissent 19 avec le circuit sécurisé 13 de façon à ce que la signature 18 de l'application contrôlée 11 soit transmise à une autre

application de vérification 24 hébergée sur la carte à puce 13, considérée ici comme un environnement sécurisé. L'application de vérification 24 dans la carte à puce maintient pour chaque type de processeur et pour chaque application à contrôler, une table de correspondance entre les instructions de contrôle exécutables 15 et les résultats attendus.

Cette table de correspondance permet de vérifier 23 la validité de la signature calculée par le certificat exécutable 15. Si le résultat de la vérification est négatif, l'application de vérification hébergée dans la carte à puce interagit 12 avec l'application à contrôler afin de modifier le fonctionnement de cette dernière. Si la vérification est positive, la carte à puce 13 produit les données 20 dont l'application logicielle 11 a besoin pour fonctionner 21.

Dans un mode de réalisation, les certificats hébergés dans la carte à puce ou le circuit sécurisé changent selon une cadence ou condition choisie.

REVENDICATIONS

1. Procédé de vérification de l'intégrité d'une application logicielle exécutable dans un terminal hôte, ***caractérisé en ce qu'***il comprend les étapes suivantes :

i) déterminer au moins une suite d'instructions de contrôle formant certificat exécutable (4,15) pour l'application logicielle, exécutable par ledit terminal hôte au cours de l'exécution de l'application logicielle à vérifier (1,11),

ii) sur le terminal hôte, exécuter l'application logicielle à vérifier (1,11), recevoir le certificat exécutable (4,15) ainsi déterminé lors de l'étape i), et exécuter la suite d'instructions de contrôle dudit certificat exécutable dans le contexte mémoire dudit terminal hôte,

iii) comparer le résultat ainsi obtenu par l'exécution des instructions de contrôle avec le résultat attendu d'une application logicielle authentique et,

iv) en cas de comparaison positive, continuer le cours de l'exécution de l'application logicielle à vérifier (1,11).

2. Procédé selon la revendication 1, dans lequel le terminal hôte est équipé d'un processeur ***caractérisé en ce que*** la suite d'instructions de contrôle formant certificat (4, 15) est codée en langage interprétable par ledit processeur du terminal hôte.

3. Procédé selon la revendication 1, dans lequel le terminal hôte est équipé d'une machine virtuelle apte à émuler un processeur, ***caractérisé en ce que*** la suite d'instructions de contrôle formant certificat (4, 15) est codée en langage interprétable par la machine virtuelle du terminal hôte.

4. Procédé selon l'une des revendications 1 à 3, dans lequel le certificat exécutable comporte une partie des traitements nécessaire au bon fonctionnement de l'application authentique.

5. Procédé selon l'une des revendications 1 à 4, ***caractérisé en ce que*** dans l'étape i) il est prévu d'établir, dans un environnement sécurisé, une carte du contexte mémoire de l'application logicielle authentique en cours d'exécution, et de déterminer, à partir des valeurs de cette carte mémoire, la suite d'instructions de contrôle destinée à former le certificat exécutable (4,15).

6. Procédé selon l'une des revendications 1 à 5, ***caractérisé en ce que*** dans l'étape ii), le certificat exécutable (4, 15) à destination du terminal hôte émane d'un circuit électronique de

traitement physiquement séparé du terminal hôte.

7. Procédé selon l'une des revendications 1 à 6, ***caractérisé en ce que*** dans l'étape ii) la récupération des valeurs du contexte mémoire d'exécution se fait par lecture des valeurs aux adresses des différentes zones de la mémoire du terminal hôte, ces zones contenant les instructions exécutables et les données intrinsèques à l'application à vérifier.

8. Procédé selon l'une des revendications 1 à 7, ***caractérisé en ce que*** dans l'étape iii), le résultat obtenu par l'exécution de ladite suite d'instructions de contrôle (4,15) produit une signature de l'application à vérifier, cette signature étant calculée par ladite suite d'instructions de contrôle (4, 15) qui utilise les valeurs du contexte mémoire de l'application logicielle à vérifier en cours d'exécution de l'application.

9. Procédé selon l'une des revendications précédentes, ***caractérisé en ce que*** l'application logicielle comprend des instructions permettant de charger et d'exécuter dans sa carte de contexte mémoire ladite suite d'instructions de contrôle (4, 15) en substituant au moins une adresse d'exécution d'une instruction de ladite application logicielle par au moins une adresse d'instruction de la suite d'instructions formant certificat.

10. Procédé selon l'une des revendications précédentes, ***caractérisé en ce que*** la suite d'instructions de contrôle (4, 15) est choisie de telle sorte que l'état du contexte mémoire d'une l'application logicielle après l'exécution de la suite d'instructions de contrôle est identique et/ou sans modification de l'état du contexte mémoire de l'application logicielle avant l'exécution de la suite d'instructions de contrôle.

11. Procédé selon l'une quelconque des revendications 1 à 10, ***caractérisé en ce que*** la suite d'instructions formant certificat (4,15) est transportée dans un flux de données nécessaire à l'exécution de l'application logicielle à vérifier.

12. Procédé selon l'une quelconque des revendications 1 à 11, ***caractérisé en ce que*** l'application logicielle à vérifier est tout ou partie chiffrée, le déchiffrement correct de l'application logicielle étant réalise en cas d'intégrité de l'application logicielle à vérifier.

13. Dispositif de vérification de l'intégrité d'une application logicielle destinée à être exécutée dans un terminal hôte pour la mise en œuvre du procédé selon l'une des revendications 1 à 12, ***caractérisé en ce qu'***il comprend des moyens de traitement aptes à déterminer au moins une suite d'instructions de contrôle (4,15) pour l'application logicielle (1,11), exécutable par ledit

terminal hôte au cours de l'exécution de l'application logicielle, et formant un certificat exécutable de ladite application logicielle, des moyens d'exécution pour exécuter la suite d'instructions formant certificat (4,15) sur le terminal hôte au cours de l'exécution de l'application logicielle, des moyens de comparaison pour comparer le résultat ainsi obtenu par l'exécution des instructions de contrôle avec le résultat attendu d'une application authentique, et des moyens aptes en cas de comparaison positive à continuer l'exécution de l'application logicielle à vérifier (1,11).

14. Dispositif selon la revendication 13, ***caractérisé en ce qu***'il comprend une carte à puce ou tout autre circuit sécurisé apte à contenir la suite d'instructions de contrôle formant certificat (4,15), en ce que le terminal hôte est équipé d'un lecteur de carte à puce ou d'un moyen de communication avec le circuit sécurisé et en ce que les moyens d'exécution de l'application logicielle sont agencés pour aller chercher, dans la carte à puce ou le circuit sécurisé, la suite d'instructions formant certificat au cours de l'exécution de l'application logicielle à vérifier.

15. Dispositif selon la revendication 14, ***caractérisé en ce que*** le terminal hôte est apte à renvoyer à la carte à puce ou au circuit sécurisé la signature produite par la suite d'instructions de contrôle, et en ce que la carte à puce ou le circuit sécurisé comprend en outre une application logicielle de vérification apte à valider ou invalider l'authenticité de l'application logicielle à vérifier en fonction du résultat de la comparaison entre la signature produite par la suite d'instructions de contrôle et une valeur de la signature connue et préalablement stockée dans la carte à puce ou dans le circuit sécurisé.

16. Dispositif selon la revendication 15, ***caractérisé en ce qu***'en cas de comparaison négative, la carte à puce est apte à modifier le fonctionnement de l'application logicielle à vérifier.

17. Dispositif selon la revendication 15 ou la revendication 16, ***caractérisé en ce qu***'en cas de non transmission de la signature conformément à des conditions prédéterminées, la carte à puce est apte à modifier le fonctionnement de l'application logicielle à vérifier.

18. Dispositif selon l'une des revendications 13 à 17, ***caractérisé en ce qu***'en cas de comparaison négative, le dispositif comprend en outre des moyens aptes à empêcher le fonctionnement de l'application logicielle dans le terminal hôte.

19. Dispositif selon l'une des revendications 13 à 18, ***caractérisé en ce que*** le terminal hôte appartient au groupe formé par les dispositifs de traitement des données, les décodeurs de télévision numérique, les équipements de visualisation de contenus multimédias, les micro-

ordinateurs, les cartes à puces, les assistants personnels, les consoles de jeux, les téléphones mobiles ou analogues.

20. Dispositif selon l'une des revendications 13 à 19, ***caractérisé en ce que*** les moyens de traitement sont aptes à déterminer une pluralité de certificats exécutables (4, 15), différents les un par rapport aux autres selon une cadence et/ou condition choisie.

21. Dispositif selon l'une des revendications 13 à 20, ***caractérisé en ce que*** les moyens de traitement sont aptes à déterminer une pluralité de certificats exécutables (14, 15), différents les uns par rapport aux autres selon une cadence et/ou une condition choisie.

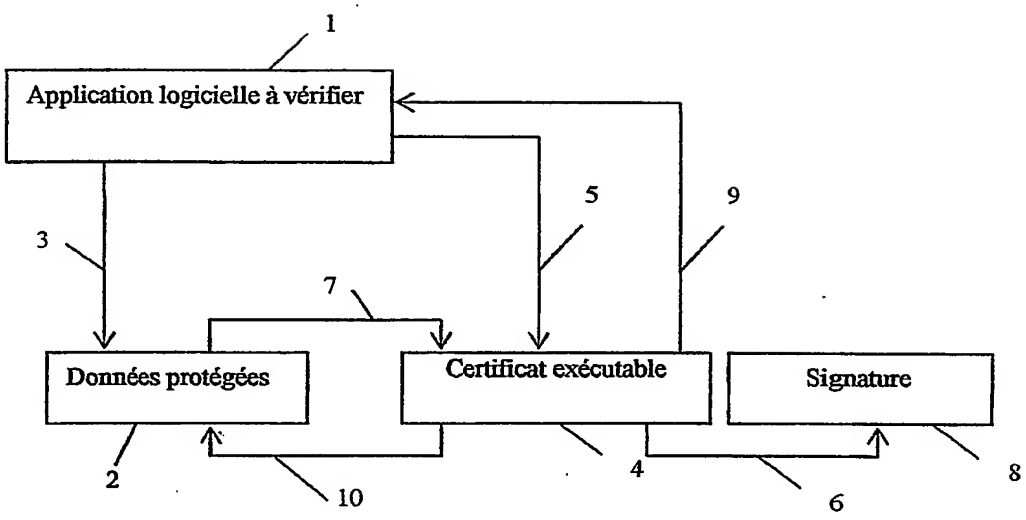


Figure 1

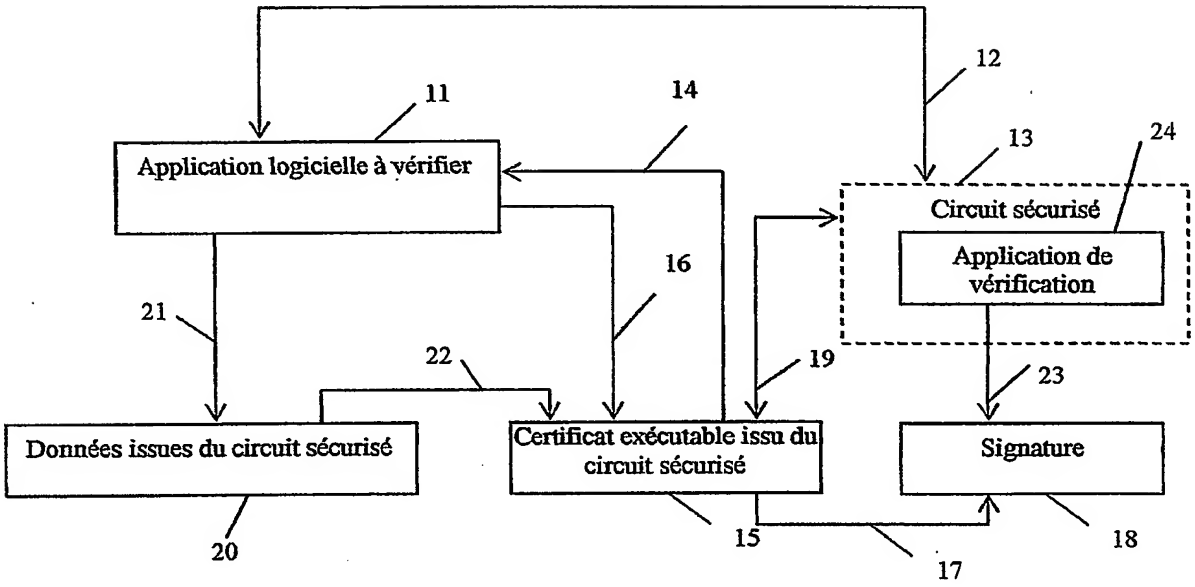


Figure 2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/FR 03/03877

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 308 270 B1 (GUTHERY SCOTT B) 23 October 2001 (2001-10-23) abstract column 1, line 33 - column 9, line 44 figures 1-5	13,18-21
Y	column 4, line 35 - line 37 column 6, line 1 - line 10 column 7, line 15 - line 33	1-12, 14-17
Y	WO 99/35582 A (LUI CHEW WAH) 15 July 1999 (1999-07-15) page 9, line 21 - page 10, line 3	1-12, 14-17

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 June 2004	16/06/2004

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Segura, G

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/FR 03/03877

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HOI CHANG AND MIKHAIL J. ATTALLAH: "Protecting Software Code by Guards" LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER VERLAG, vol. 2320, 10 June 2002 (2002-06-10), pages 160-175, XP002245264 ISBN: 3540436774 page 162 - page 163 page 168	13,18-21
A	US 6 006 328 A (DRAKE CHRISTOPHER NATHAN) 21 December 1999 (1999-12-21) column 5, line 62 - column 6, line 3	12
A	WO 00/33196 A (MUIR ROBERT ;LYONS MARTIN (AU); ARISTOCRAT LEISURE IND PTY LTD (AU) 8 June 2000 (2000-06-08) the whole document	1-21

INTERNATIONAL SEARCH REPORT

International Application No
PCT/FR 03/03877

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6308270 B1	23-10-2001	NONE	
WO 9935582 A	15-07-1999	AU 8251998 A	26-07-1999
		WO 9935582 A1	15-07-1999
US 6006328 A	21-12-1999	AU 725098 B2	05-10-2000
		AU 5945796 A	23-01-1997
		WO 9704394 A1	06-02-1997
WO 0033196 A	08-06-2000	AU 767422 B2	06-11-2003
		AU 1539300 A	19-06-2000
		WO 0033196 A1	08-06-2000
		US 6722986 B1	20-04-2004

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No
PCT/FR 03/03877

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 G06F1/00

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 7 G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	US 6 308 270 B1 (GUTHERY SCOTT B) 23 octobre 2001 (2001-10-23) abrégé colonne 1, ligne 33 - colonne 9, ligne 44 figures 1-5	13,18-21
Y	colonne 4, ligne 35 - ligne 37 colonne 6, ligne 1 - ligne 10 colonne 7, ligne 15 - ligne 33	1-12, 14-17
Y	WO 99/35582 A (LUI CHEW WAH) 15 juillet 1999 (1999-07-15) page 9, ligne 21 - page 10, ligne 3	1-12, 14-17

-/--

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

° Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'inven tion revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'inven tion revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
8 juin 2004	16/06/2004

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Segura, G

Formulaire PCT/ISA/210 (deuxième feuille) (Janvier 2004)

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	HOI CHANG AND MIKHAIL J. ATTALLAH: "Protecting Software Code by Guards" LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER VERLAG, vol. 2320, 10 juin 2002 (2002-06-10), pages 160-175, XP002245264 ISBN: 3540436774 page 162 - page 163 page 168	13,18-21
A	US 6 006 328 A (DRAKE CHRISTOPHER NATHAN) 21 décembre 1999 (1999-12-21) colonne 5, ligne 62 - colonne 6, ligne 3	12
A	WO 00/33196 A (MUIR ROBERT ;LYONS MARTIN (AU); ARISTOCRAT LEISURE IND PTY LTD (AU) 8 juin 2000 (2000-06-08) le document en entier	1-21

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No
PCT/FR 03/03877

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de famille de brevet(s)		Date de publication
US 6308270	B1	23-10-2001	AUCUN		
WO 9935582	A	15-07-1999	AU	8251998 A	26-07-1999
			WO	9935582 A1	15-07-1999
US 6006328	A	21-12-1999	AU	725098 B2	05-10-2000
			AU	5945796 A	23-01-1997
			WO	9704394 A1	06-02-1997
WO 0033196	A	08-06-2000	AU	767422 B2	06-11-2003
			AU	1539300 A	19-06-2000
			WO	0033196 A1	08-06-2000
			US	6722986 B1	20-04-2004